第 23 卷 第 12 期 控 制 与 决 策 2008 年 12 月
Vol. 23 No. 12 Control and Decision Dec. 2008

文章编号: 1001-0920(2008)12-1413-04

一类具有资源约束和恶化效应的单机成组排序问题

闫 杨, 王大志, 汪定伟, 王洪峰
(东北大学 信息科学与工程学院, 沈阳 110004)

摘 要: 讨论具有连续资源的单机成组排序问题. 这一模型中同一组内的工件不允许分开加工,各工件组的安装时间是所消耗资源的非负减少连续函数,工件的加工时间是开工时间的严格减少函数. 针对满足资源消耗总量限制条件下极小化最大完工时间的问题,以及在满足最大完工时间限制条件下极小化资源消耗总量的问题,讨论了最优排序的某些特征,分别给出了求解最优资源分配的方法. 最后通过数值例子表明了所提出方法的正确性和有效性.

关键词: 单机排序; 成组技术; 资源约束; 安装时间
中图分类号: O223 **文献标识码**: A

Single machine group scheduling problem with resource constraints and deteriorating jobs

YAN Yang, WANG Da-zhi, WANG Ding-wei, WANG Hong-feng
(School of Information Science and Engineering, Northeastern University, Shenyang 110004, China. Correspondent: YAN Yang, E-mail: yanyangmail@163.com)

Abstract: The single machine group scheduling with continuous resources is discussed. In this model, the jobs in the same group shouldn't be separated, the setup time of a group is a positive decreasing function of the amount of resources consumed, and the processing time of a job is a strictly decreasing linear function of its starting time. For the total resource consumption minimization problem under the makespan constrains, and the makespan minimization problem under the total resource consumption constrains, the characterizations of optimal schedules are discussed, and the optimal allocation methods are presented respectively. We also illustrate them by examples.

Key words: Single-machine scheduling; Group technology; Resource constrained; Setup time

1 引 言

在经典的排序问题中,通常假设工件的加工时间是常数,且不考虑资源消耗以及成组技术限制. 但在某些具有较强实际背景的排序问题中,这些假设可能并不满足. 对于工件加工时间是其开工时间函数的一类问题,迄今已有广泛的应用,并取得了较多的研究结果[1-3],有关这方面的综述可参见文献[4]. Ho 等[5]提出了工件加工时间线性减少的模型,该模型在现实生活中具有广泛的应用背景. 例如:工人在加工相同或相似工件时,随着时间的增长,操作技能提高,能够更好地调节组织各项资源和设备,因而工件所需的加工时间逐渐减少;雷达识别空中飞行威胁物体的过程中,若雷达系统已经发现了正在向其接近的目标,则随着目标的靠近,雷达系统识别目标所需的时间越来越短,因而目标被发现的越晚,所需的识别时间越短[5]. 在成组排序问题中,工件可以分成"类似"工件的工件组,同组工件连续加工时不需要安装时间,而不同组工件连续加工时则需要一定的安装时间[6,7]. 但在实际问题中,安装时间往往受到某些资源(如机械加工中特殊的润滑剂,化工产品的催化剂等)的影响,工件组的安装时间会相应减少[8]. 对于工件的准备时间为资源消耗量函数的情况,文献[9-11]进行了讨论,但对于安装时间与资源有关的一类单机成组排序问题的结论并不多[12,13].

本文将讨论更一般的情况,即工件组的安装时间为所消耗资源的非负严格减少连续函数,工件的加工时间是开工时间的严格减少线性函数的单机成组排序问题.

收稿日期: 2007-08-18; **修回日期**: 2007-10-23.
基金项目: 国家自然科学基金重点项目(70431003); 国家自然科学基金创新群体项目(60521003); 国家科技支撑计划项目(2006BAH02A09).
作者简介: 闫杨(1982—),女,辽宁抚顺人,博士生,从事生产调度理论、动态进化算法的研究; 汪定伟(1948—),男,江西彭泽人,教授,博士生导师,从事模糊建模与智能优化算法、生产调度理论等研究.

首先给出问题的一般描述:设有 n 个工件 J_1, J_2, $\cdots J_n$,在一台机器上加工,若工件 J_j 的开工时间为 t,则其加工时间为 $p_j(a - bt)$,其中 a 和 b 为非负常数, $j = 1,2,\cdots,n$. 不失一般性,设所有工件的到达时间 $t_0 = 0$. n 个工件分别属于 m 组 $G_1, G_2, \cdots, G_m$,第 i 组 G_i 包含 $n_i\left(\sum_{i=1}^{m} n_i = n\right)$ 工件 $J_{i1}, J_{i2}, \cdots, J_{in_i}$,工件 J_{ij} 的基本加工时间为 p_{ij}. 同组工件必须接连放在一起加工,加工第 i 组工件时所需安装时间为 $s_i = f(u_i)$, $0 \le u_i \le \bar{u}$, $i = 1,2,\cdots,m$. 其中: u_i 为分配给工件组 G_i 的不可再生资源量; $\bar{u}$ 为资源分配量的上限; f 为非负严格减少连续函数,即 $f(0) > f(\bar{u}) \ge 0$,且反函数 f^{-1} 可以求出.

下面讨论两类问题:第 1 类问题是在满足资源消耗总量限制条件下,极小化最大完工时间;第 2 类问题是在满足最大完工时间限制条件下,极小化资源消耗总量. 用三参数表示法将问题分别记为

$$1 \mid s_i = f(u_i), G, p_{ij}(a - bt), \sum u_i \le U \mid C_{\max}; \tag{1}$$

$$1 \mid s_i = f(u_i), G, p_{ij}(a - bt), C_{\max} \le C \mid \sum u_i. \tag{2}$$

2 极小化最大完工时间问题

引理 1 问题 $1 \mid p_j(a - bt) \mid C_{\max}$ 的最大完工时间与工件加工顺序无关,若第 1 个加工的工件开工时间为 t_0,则最大完工时间为

$$C_{\max}(t_0 \mid J_1, J_2, \cdots, J_n) = \left(t_0 - \frac{a}{b}\right)\prod_{j=1}^{n}(1 - bp_j) + \frac{a}{b}.$$

定理 1 对于问题 $1 \mid G, p_{ij}(a - bt) \mid C_{\max}$,若第 1 个加工的组开工时间为 $t_0 = 0$,则最大完工时间为

$$C_{\max} = \sum_{k=1}^{m}\left(s_k \prod_{i=k}^{m}\prod_{j=1}^{n_i}(1 - bp_{ij})\right) - \frac{a}{b}\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1 - bp_{ij}) + \frac{a}{b}. \tag{3}$$

证明 记 $\pi = [G_{\pi(1)}, G_{\pi(2)}, \cdots, G_{\pi(m)}] = [G_1, G_2, \cdots, G_m]$ 为工件组的一个调度,各工件在 $t_0 = 0$ 时刻到达,设工件组 G_i 中有 n_i 个工件, $i = 1,2,\cdots,m$,第 i 组的第 j 个工件记为 J_{ij},则第 1 组和第 2 组的完工时间分别为

$$C_1 = \left(s_1 - \frac{a}{b}\right)\prod_{j=1}^{n_1}(1 - bp_{1j}) + \frac{a}{b},$$

$$C_2 = \left(C_1 + s_2 - \frac{a}{b}\right)\prod_{j=1}^{n_2}(1 - bp_{2j}) + \frac{a}{b} = \sum_{k=1}^{2}\left(s_k \prod_{i=k}^{2}\prod_{j=1}^{n_i}(1 - bp_{ij})\right) - \frac{a}{b}\prod_{i=1}^{2}\prod_{j=1}^{n_i}(1 - bp_{ij}) + \frac{a}{b}.$$

设对于第 l 组 G_l,式(3) 成立,有

$$C_l = \sum_{k=1}^{l}\left(s_k \prod_{i=k}^{l}\prod_{j=1}^{n_i}(1 - bp_{ij})\right) - \frac{a}{b}\prod_{i=1}^{l}\prod_{j=1}^{n_i}(1 - bp_{ij}) + \frac{a}{b}.$$

考虑第 $l+1$ 组 G_{l+1},有

$$\begin{aligned} C_{l+1} ={}& \left(C_l + s_{l+1} - \frac{a}{b}\right)\prod_{j=1}^{n_{l+1}}(1 - bp_{l+1,j}) + \frac{a}{b} = \\ & \Big[\sum_{k=1}^{l}\left(s_k \prod_{i=k}^{l}\prod_{j=1}^{n_i}(1 - bp_{ij})\right) - \frac{a}{b}\prod_{i=1}^{l}\prod_{j=1}^{n_i}(1 - bp_{ij}) + \frac{a}{b} + s_{l+1} - \\ & \frac{a}{b}\Big]\prod_{j=1}^{n_{l+1}}(1 - bp_{l+1,j}) + \frac{a}{b} = \\ & \sum_{k=1}^{l+1}\left(s_k \prod_{i=k}^{l+1}\prod_{j=1}^{n_i}(1 - bp_{ij})\right) - \frac{a}{b}\prod_{i=1}^{l+1}\prod_{j=1}^{n_i}(1 - bp_{ij}) + \frac{a}{b}, \end{aligned}$$

即对于第 $l+1$ 组 G_{l+1},式(3) 成立. 由归纳假设,对于第 m 组 G_m,式(3) 成立. □

性质 1 最大完工时间不受各组内工件加工次序的影响.

证明 在最大完工时间的表达式(3) 中,设

$$K = \frac{a}{b} - \frac{a}{b}\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1 - bp_{ij}),$$

则 K 是与次序无关的常数. 而 $\sum_{k=1}^{m}\left(s_k \prod_{i=k}^{m}\prod_{j=1}^{n_i}(1 - bp_{ij})\right)$ 不受各组内工件加工次序的影响. □

性质 2 对于工件组间的任意一个调度,都应优先将资源分配给排位在后的组,才可以极小化目标函数 $C_{\max}$.

证明 不失一般性,设 $\pi = [G_1, G_2, \cdots, G_m]$ 为工件组间的任意一个调度. 由最大完工时间的表达式(3) 可以得出,应优先给 $\prod_{i=l}^{m}\prod_{j=1}^{n_i}(1 - bp_{ij})$ 大的安装时间 s_l 分配资源,以相同资源量可以极小化最大完工时间. 因为 $b > 0$, $p_{ij} > 0$, $i = 1,2,\cdots,m$, $j = 1,2,\cdots,n_i$,所以对于任何 i 都有

$$\prod_{j=1}^{n_i}(1 - bp_{ij}) < 1,$$

$$\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1-bp_{ij}) <$$

$$\prod_{i=2}^{m}\prod_{j=1}^{n_i}(1-bp_{ij}) < \cdots < \prod_{i=m}^{m}\prod_{j=1}^{n_i}(1-bp_{ij}).$$

因此,应对排位在后的组优先分配资源,从而可以极小化目标函数 $C_{\max}$. □

性质 3　按性质 2,可为各个位置上的工件组分配资源. 当对每个位置上工件组的资源分配量一定时,应按 $\rho_{G_i} = \prod_{j=1}^{n_i}(1-bp_{ij})$ 不增顺序为工件组调度 $(i=1,2,\cdots,m)$,所得的调度为最优调度.

证明　不失一般性,将为排在第 i 个位置的组分配的资源量记为 $u^*_{[i]}$, $i=1,2,\cdots,m$,假设最优调度为 $\pi=[G_1,G_2,\cdots,G_m]$. 在 π 中, G_l 与 G_{l+1} 为两相邻工件组(G_l 排在 G_{l+1} 前),且有

$$\prod_{j=1}^{n_l}(1-bp_{lj}) < \prod_{j=1}^{n_{l+1}}(1-bp_{l+1,j}).$$

此时,若交换 G_l 与 G_{l+1} 的顺序,则可得到一个新调度 π'. π 与 π' 所对应的目标函数值分别为

$$C_{\max} = C_m =$$
$$f(u^*_{[1]})\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1-bp_{ij}) + \cdots + f(u^*_{[m]})\prod_{j=1}^{n_m}(1-bp_{mj}) - \frac{a}{b}\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1-bp_{ij}) + \frac{a}{b} =$$
$$\sum_{k=1}^{m} f(u^*_{[k]})\prod_{k=l}^{m}\prod_{j=1}^{n_i}(1-bp_{ij}) + K,$$

$$C'_{\max} = C'_m =$$
$$f(u^*_{[1]})\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1-bp_{ij}) + \cdots +$$
$$f(u^*_{[l-1]})\prod_{i=l-1}^{m}\prod_{j=1}^{n_i}(1-bp_{ij}) + f(u^*_{[l]})\prod_{j=1}^{n_{l+1}}(1-bp_{l+1,j})\prod_{j=1}^{n_l}(1-bp_{lj})\prod_{i=l+2}^{m}\prod_{j=1}^{n_i}(1-bp_{ij}) +$$
$$f(u^*_{[l+1]})\prod_{j=1}^{n_l}(1-bp_{l,j})\prod_{i=l+2}^{m}\prod_{j=1}^{n_i}(1-bp_{ij}) +$$
$$f(u^*_{[l+2]})\prod_{i=l+2}^{m}\prod_{j=1}^{n_i}(1-bp_{ij}) + \cdots + f(u^*_{[m]})\prod_{j=1}^{n_m}(1-bp_{ij}) - \frac{a}{b}\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1-bp_{ij}) + \frac{a}{b} =$$
$$\prod_{k=1}^{l-1} f(u^*_{[k]})\prod_{k=l}^{m}\prod_{j=1}^{n_i}(1-bp_{ij}) +$$
$$f(u^*_{[l]})\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1-bp_{ij}) +$$
$$f(u^*_{[l+1]})\prod_{j=1}^{n_l}(1-bp_{l,j})\prod_{i=l+2}^{m}\prod_{j=1}^{n_i}(1-bp_{ij}) +$$
$$\sum_{k=l+2}^{m} f(u^*_{[k]})\prod_{k=l}^{m}\prod_{j=1}^{n_i}(1-bp_{ij}) + K,$$

其中 $K = \frac{a}{b} - \frac{a}{b}\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1-bp_{ij})$. 因而可得到

$$C_{\max} - C'_{\max} =$$
$$\prod_{i=l+2}^{m}\prod_{j=1}^{n_i}(1-bp_{ij}) f(u^*_{[l+1]})\Big[\prod_{j=1}^{n_{l+1}}(1-bp_{l+1,j}) - \prod_{j=1}^{n_l}(1-bp_{lj})\Big] > 0.$$

因此,调度 π' 优于调度 π,这与 π 为最优调度矛盾. 若为排在每个位置上工件组的资源分配量一定时,则按 $\rho_{G_i} = \prod_{j=1}^{n_i}(1-bp_{ij})$ 不增顺序为工件组调度 $(i=1,2,\cdots,m)$,所得的调度为最优调度. □

性质 4　在资源总量受限前提下,在 $\sum_{i=1}^{m}u_i = U$ 时极小化最大完工时间,可使最大完工时间最小.

根据上述分析,可得到问题 $1\,|\,s_i = f(u_i), G, p_{ij}(a-bt), \sum u_i \le U\,|\,C_{\max}$ 的最优算法.

算法 1(复杂性为 $O(n\log n)$)

Step1: 将各组按 $\rho_{G_i} = \prod_{j=1}^{n_i}(1-bp_{ij})$ 不增顺序排序 $(i=1,2,\cdots,m)$,得 $\pi=[G_{[1]},G_{[2]},\cdots,G_m]$,置 $u^*_{[i]}=0$, $i=1,2,\cdots,m$, $k=m$.

Step2: 置 $u^*_{[k]}=\min\{\bar{u},U\}$, $U=U-u^*_{[k]}$, $k=k-1$.

Step3: 若 $U=0$ 或 $k=0$,则算法停止;否则转 Step2.

3　极小化资源消耗总量问题

通过上述对问题(1)及性质 1～性质 4 的分析可知,对于问题 $1\,|\,s_i = f(u_i), G, p_{ij}(a-bt), C_{\max} \le C\,|\,\sum u_i$,首先资源消耗总量 $\sum u_i$ 不受各工件组内工件加工次序的影响;其次,为了极小化资源消耗总量,仍需将各工件组按 $\rho(G_i) = \prod_{j=1}^{n_i}(1-bp_{ij})$ 不增的顺序对各组进行调度,所得的调度为最优调度,而且在 $C_{\max} = C$ 时极小化资源消耗总量,可使资源消耗总量最小,并应对排位在后的组优先分配资源. 由 $C_{\max}$ 的表达式可知,若只对工件组 $G_{\pi[m]}$ 的安装时间分配资源,而其余组的资源分配量均为 0 时,有

$$s\sum_{k=1}^{m-1}\prod_{i=1}^{k}\prod_{j=1}^{n_{[i]}}(1-bp_{[i]j}) +$$

$$s_{[m]}\prod_{i=1}^{m}\prod_{j=1}^{n_{[i]}}(1-bp_{[i]j})=C-K,$$

则对组 $G_{\pi[m]}$ 的安装时间应为

$$s_{[m]}=C-K-\frac{s\sum_{k=1}^{m-1}\prod_{i=1}^{k}\prod_{j=1}^{n_{[i]}}(1-bp_{[i]j})}{\prod_{i=1}^{m}\prod_{j=1}^{n_{[i]}}(1-bp_{[i]j})}.$$

其中

$$s=f(0),\ K=\frac{a}{b}-\frac{a}{b}\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1-bp_{ij}).$$

但对组 $G_{[m]}$ 所分得的资源量受到限制,即 $0\leq u^*_{[m]}\leq\bar{u}$. 因此,若 $s_{[m]}\geq f(0)$,则无需对组 $G_{[m]}$ 分配资源,即 $u^*_{[m]}=0$ 便可满足 $C_{\max}\leq C$,从而对排列在前的各组也无需分配资源,算法停止;若 $f(\bar{u})\leq s_{[m]}<f(0)$,则对组 $G_{[m]}$ 分配资源 $u^*_{[m]}\in(0,\bar{u}]$,此时求得 $u^*_{[m]}=f^{-1}(s_{[m]})$,无需为排列在之前的各组分配资源,算法停止;若 $s_{[m]}<f(\bar{u})$,则对组 $G_{[m]}$ 的资源分配量为 $\bar{u}$,即 $u^*_{[m]}=\bar{u}$,并且对排列在前的组仍需分配资源. 此时已对 $G_{[m]}$ 分配资源量达到上限,按同样的方法可依次计算出 $u^*_{[m-1]},\cdots,u^*_{[1]}$.

下面给出问题 $1\mid s_i=f(u_i),G,p_{ij}(a-bt),C_{\max}\leq C\mid\sum u_i$ 的最优算法.

算法 2 (复杂性为 $\max\{O(n\log n),O(ng(n))\}$,其中 $g(n)$ 为 f^{-1} 的复杂性)

Step1: 将各组按 $\rho(G_i)=\prod_{j=1}^{n_i}(1-bp_{ij})$ 不增顺序排序 $(i=1,2,\cdots,m)$,得 $\pi=[G_{[1]},G_{[2]},\cdots,G_m]$.

Step2: 置 $u^*_{[i]}=0$, $i=1,2,\cdots,m$, $U=0$, $s=f(0)$, $s'=f(\bar{u})$, $l=m$,计算

$$K=\frac{a}{b}-\frac{a}{b}\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1-bp_{ij}),$$

置 $C'=C-K$.

Step3: 计算

$$s_{[l]}=C'-s\frac{\sum_{k=1}^{l-1}\prod_{i=1}^{k}\prod_{j=1}^{n_{[i]}}(1-bp_{[i]j})}{\prod_{i=1}^{l}\prod_{j=1}^{n_{[i]}}(1-bp_{[i]j})}.$$

若 $s_{[l]}\geq f(0)$,则算法终止;若 $f(\bar{u})\leq s_{[l]}<f(0)$,则置 $u^*_{[l]}=f^{-1}(s_{[l]})$, $U=U+u^*_{[l]}$,算法终止,最优资源分配为 $u^*=[u_1^*,u_2^*,\cdots,u_m^*]$;若 $s_{[l]}<f(\bar{u})$,则转 Step4.

Step4: 置

$u^*_{[l]}=\bar{u}$, $U=U+u^*_{[l]}$,

$$C'=C'-s'\prod_{i=l}^{m}\prod_{j=1}^{n_i}(1-bp_{ij}),\ l=l-1.$$

若 $l<0$,则算法终止,问题无解;否则转 Step3.

4 数值例子

求解问题

$$1\mid s_i=f(u_i),G,p_{ij}(a-bt),\ \sum u_i\leq U\mid C_{\max}.$$

其中: $n=10$, $m=4$, $G_1=\{J_1,J_2,J_3\}$, $G_2=\{J_4,J_5,J_6\}$, $G_3=\{J_7,J_8\}$, $G_4=\{J_9,J_{10}\}$, $p_1=0.2$, $p_2=0.1$, $p_3=0.1$, $p_4=0.2$, $p_5=0.1$, $p_6=0.2$, $p_7=0.3$, $p_8=0.3$, $p_9=0.4$, $p_{10}=0.4$. 设 $a=20$, $b=1$, $s_i=f(u_i)$ 为线性函数, $s_i=r_0-r_1u_i$, $r_0=30$, $r_1=5$, $\bar{u}=5$, $U=14$.

首先计算出 $\rho_1=0.648$, $\rho_2=0.576$, $\rho_3=0.49$, $\rho_4=0.42$,工件组间排序为 $\pi=[G_1,G_2,G_3,G_4]$. $i=4$, $u_4^*=\min\{5,14\}=5$, $U=14-5=9$; $i=3$, $u_3^*=\min\{5,9\}=5$, $U=9-5=4$; $i=2$, $u_2^*=\min\{5,4\}=4$, $U=4-4=0$; $u_1^*=0$, $\pi=[G_1,G_2,G_3,G_4]$, $u^*=[u_1^*,u_2^*,u_3^*,u_4^*]=[0,4,5,5]$. 因而 $C_{\max}=26$.

5 结 论

对于具有资源约束和恶化效应单机成组排序问题,目前的结论并不完备. 为此,本文引入一类安装时间受资源约束,工件加工时间随开工时间线性减少的单机成组调度模型. 给出了问题 $1\mid s_i=f(u_i),G,p_{ij}(a-bt),\ \sum u_i\leq U\mid C_{\max}$ 和 $1\mid s_i=f(u_i),G,p_{ij}(a-bt),C_{\max}\leq C\mid\sum u_i$ 的若干性质以及最优算法,并通过数值例子验证了算法的正确性和有效性. 但是,对于受资源约束成组问题的其他模型,还有待进一步研究.

参考文献(References)

[1] Mosheiov G. V-shaped policies for scheduling deteriorating jobs[J]. Operations Research, 1991, 39(6): 979-991.

[2] Mosheiov G. Scheduling jobs under simple linear deterioration[J]. Computer and Operational Research, 1994, 21(6): 653-659.

[3] Wang J B, Wang M, Xia Z Q. Single-machine scheduling to minimize makespan under linear deterioration [J]. Int J of Pure and Applied Mathematics, 2004, 13(4): 529-539.

[4] Cheng T C E, Ding Q, Lin B M T. A concise survey of scheduling with time-dependent processing times[J]. European J of Operational Research, 2004, 152(1): 1-13.

(下转第 1422 页)

1 : 67-72.

[2] 喻寿益, 郭观七. 选择的遗传漂移分析[J]. 计算机研究与发展, 2004, 41(2): 346-351.
(Yu S Y, Guo G Q. Genetic drift analysis of selection [J]. J of Computer Research and Development, 2004, 41(2): 346-351.)

[3] Goldberg D E, Richardson J. Genetic algorithms with sharing for multimodal function optimization[C]. Proc of the 2nd Int Conf on Genetic Algorithms. NJ: Lawrence Erlbaum, 1987: 41-49.

[4] Mahfoud S W. Crowding and preselection revisited[C]. Parallel Problem Solving from Nature. Amsterdam: Elsevier, 1992: 27-36.

[5] Li J P, Balazs M E, Parks G T. A species conserving genetic algorithm for multimodal function optimization [J]. J of Evolutionary Computation, 2002, 10(3): 207-234.

[6] 黄席樾. 现代智能算法理论及应用[M]. 北京: 科学出版社, 2005.
(Huang X Y. Modern intelligent algorithms theory and applications[M]. Beijing: Science Press, 2005.)

[7] De Castro L N, Von Zuben F J. Learning and optimization using the clonal selection principle [J]. IEEE Trans on Evolutionary Computation, 2002, 6(3): 239-251.

[8] De Castro L N, Timmis J. An artificial immune network for multimodal function optimization[C]. Proc of IEEE Conf on Evolutionary Computation. Hawaii, 2002, 1: 699-704.

[9] 罗印升, 李人厚, 张维玺. 基于免疫机理的多峰值函数并行优化算法[J]. 系统仿真学报, 2005, 17(2): 319-322.
(Luo Y S, Li R H, Zhang W X. Multimodal functions parallel optimization algorithm based on immune mechanism[J]. J of System Simulation, 2005, 17(2): 319-322.)

[10] 王向军, 向东. 一种双种群进化规划算法[J]. 计算机学报, 2006, 29(5): 835-840.
(Wang X J, Xiang D. A novel bi-group evolutionary programming[J]. Chinese J of Computers, 2006, 29(5): 835-840.)

[11] 谭竹梅, 余晓峰, 郭观七. 排挤小生态遗传算法的改进方法[J]. 控制理论与应用, 2004, 21(4): 651-654.
(Tan Z M, Yu X F, Guo G Q. Improvement of niching genetic algorithms using crowding[J]. Control Theory and Applications, 2004, 21(4): 651-654.)

[12] 张著洪, 黄席樾. 一种新的免疫算法及其在多模态函数优化中的应用[J]. 控制理论与应用, 2004, 21(1): 17-21.
(Zhang Z H, Huang X Y. Novel immune algorithm and its application to multi-modal function optimization [J]. Control Theory and Applications, 2004, 21(1): 17-21.)

[13] Sareni B, Krahenbuhl L. Fitness sharing and niching methods revisited [J]. IEEE Trans on Evolutionary Computation, 1998, 2(3): 97-106.

(上接第 1416 页)

[5] Ho K J, Leung J T, Wei W D. Complexity of scheduling tasks with time-dependent execution times [J]. Information Processing Letters, 1993, 48: 315-320.

[6] Potts C N, Kovalyov M Y. Scheduling with batching: A review[J]. European J of Operational Research, 2000, 120(2): 228-249.

[7] Potts C N, Van Wassenhove L N. Integrating scheduling with batching and lot-sizing: A review of algorithms and complexity[J]. J of Operational Research Society, 1992, 43(3): 395-406.

[8] 张峰. 工件加工时间随工件开工时间线性增加的成组排序问题[J]. 上海第二工业大学学报, 2001, 2(2): 7- 9.
(Zhang F. Batching problem about increase of processing times[J]. J of Shanghai Second Polytechnic University, 2001, 2(2): 7-9.)

[9] Cheng T C E, Janiak A. Resource optimal control in some single-machine scheduling problems [J]. IEEE Trans on Automatic Control, 1994, 39(6): 1243-1246.

[10] Zhao C L, Tang H Y. Single machine scheduling with linear processing times [J]. Acta Automatica Sinica, 2003, 5(9): 703-708.

[11] 赵传立, 唐恒永. 一类资源约束的单机排序问题[J]. 系统工程学报, 2004, 19(5): 451-456.
(Zhao C L, Tang H Y. Single machine scheduling with resource constrains [J]. J of Systems Engineering, 2004, 19(5): 451-456.)

[12] 闫杨, 赵传立. 安装时间受资源约束的单机成组调度问题[J]. 电机与控制学报, 2007, 11(1): 70-73.
(Yan Y, Zhao C L. Single machine group scheduling with learning effect and resource dependent setup times [J]. Electric Machines and Control, 2007, 11(1): 70-73.)

[13] 闫杨, 赵传立. 一类安装时间受资源约束的单机成组排序问题[J]. 系统工程与电子技术, 2007, 29(6): 938-941.
(Yan Y, Zhao C L. Single machine group scheduling with resource dependent setup times [J]. Systems Engineering and Electronics, 2007, 29(6): 938-941.)